Migrating WebSphere Application to IBM Cloud Private Using Transformation Advisor

Overview

This document describes how to use the Migration Bundle produced by IBM Transformation Advisor to migrate your WebSphere applications to Liberty running on IBM private Cloud. There are a number of steps in the process, and each need to be completed in turn.

1. **Migrate the WebSphere application to Liberty profile**
    - You configure the Liberty profile for your application and add the necessary application and third party binaries.

2. **Containerize the Liberty profile running the application**
    - You create a docker image of your working application running in the correctly configured Liberty profile

3. **Deploy the docker image to IBM private Cloud**
    - Tag and add the image to the IBM Private Cloud Image repository. Then create a running instance of this image in IBM Private Cloud

4. **OPTIONAL: Deploy Helm Charts of the docker image to IBM Private Cloud**
    - Deploy a Helm chart to IBM Private Cloud so that you can create parameterized instances of your image, as many times as you want

Prerequisites

The prerequisites for each Step are listed at the beginning of that Step. The items listed here is the complete set of prerequisites required to run every Step

- The Migration Bundle provided by IBM Transformation Advisor
- An instance of IBM Cloud Private
- A linux machine with access to the internet with the following software installed:
    - WebSphere Application Server Liberty profile
    - Docker (download from here)
    - The Kubectl CLI
    - The Helm CLI

- The IBM Cloud CLI (download here)
- The IBM Cloud Private CLI (download here)

Step 1: Migrating to Liberty Profile

In this Step you will migrate your WebSphere Application to Liberty using the migration helper artifacts from the Transformation Advisor Migration Bundle.

Prerequisites

1. To complete this stage you require the following software to be installed:
    - WebSphere Application Server Liberty profile
2. To complete this stage you need to copy the Migration Bundle that you downloaded from Transformation Advisor to the machine where you have installed the WebSphere Application Server Liberty profile. Unzip the bundle.

You will use the following items from the Migration Bundle:

- server.xml
- All binaries

In the following tasks:

- <MIGRATION_BUNDLE_HOME> refers to the location where you have unzipped the Transformation Advisor Migration Bundle.
- Liberty profile refers to WebSphere Application Server Liberty profile.
- <LIBERTY_HOME> refers to the location where you have installed Liberty profile
- <LIBERTY_HOME_MACHINE_IP> refers to the IP address of the machine where you have installed Liberty profile
- <APP_CONTEXT> refers to the context for you applicaiton

Tasks

1. Create a server in the Liberty profile to run your application

    ```
    cd <LIBERTY_HOME>/bin
    ./server create server1
    ```

2. Move to the location of your unzipped Migration Bundle and copy the application binary (ear/war) from the Migration Bundle to the dropins directory of the WebSphere Application Server Liberty profile

   ```
   cd <MIGRATION_BUNDLE_HOME>

   cp binary/application/* <LIBERTY_HOME>/usr/servers/server1/dropins
   ```

3. Create a lib directory for any additional binaries that are required and copy them into place

   ```
   mkdir <LIBERTY_HOME>/usr/servers/server1/lib

   cp binary/lib/* <LIBERTY_HOME>/usr/servers/server1/lib
   ```

   NOTE: If you did not upload all the binary files before generating the Migration Bundle you should place them here before running the above commands

4. Update the server.xml if necessary

   NOTE: You may wish to change the port numbers given for the httpEndpoint

   NOTE: All passwords have been replaced with '???'. Enter the correct password now.

   NOTE: If there are any additional binaries listed in this file that you do not need, you should remove any reference to them.

5. Copy the server.xml into place

   ```
   cp server.xml <LIBERTY_HOME>/usr/servers/server1/server.xml
   ```

6. Start the Liberty profile server

   ```
   <LIBERTY_HOME>/bin/server start server1
   ```

7. Check the Liberty profile logs to confirm that your application has started correctly and to find the URL to access it at

   ```
   cd <LIBERTY_HOME>/usr/servers/server1/logs

   vi messages.log
   ```

   *NOTE: If you define a <dataSource> in server.xml you may encounter an authentication issue similar to this: **invalid username/password; logon denied**. If you see this issue you may need to enter your username and password in line. Do this by adding the attributes 'user' and 'password' to the property. For example like this:*

<properties.oracle portNumber="1521" URL="jdbc:oracle:thin:@9.9.9.9:1522:orcl"
user="system" password ="TransAdv01" *……/>*

8. In the logs there should be a line similar to this:

   *TCP Channel defaultHttpEndpoint has been started and is now listening for requests on host * (IPv6) port 9080*

9. Open your application in the browser by going to the following link

   http:// <LIBERTY_HOME_MACHINE_IP>:9080/<APP_CONTEXT>

Note: The Migration Bundle provides artifacts to assist you in the migration. Depending on the nature and complexity of your application additional configuration may be required to fully complete this Task.

Step 2: Containerizing Liberty

In this Step you will containerize your working Liberty profile. This will involve create an image of your Liberty profile that has your migrated application installed and working. You will then test the image and confirm that it is operating correctly.

Prerequisites

1. You have completed Step 1:Migrating To Liberty Profile
2. To complete this stage you require the following software to be installed:
    - Docker (download from here)
3. The machine where you complete this task requires access to the internet to download the Liberty base image.

You will use the following items from the Migration Bundle:

- Dockerfile
- server.xml
- All binaries

In the following tasks:

- <MIGRATION_BUNDLE_HOME> refers to the location where you have unzipped the Transformation Advisor Migration Bundle.
- <LIBERTY_HOME> refers to the location where you have installed Liberty profile
- <IMAGE_NAME> is the name you will use for this image, typically the name of the application.

Tasks

1. Stop the Liberty profile if it is running to ensure that the necessary ports are freed

    ```
    <LIBERTY_HOME>/bin/server stop server1
    ```

2. Ensure the docker service is running

    ```
    service docker start
    ```

3. Go to your Migration Bundle and build your image from the docker file

   ```
   cd <MIGRATION_BUNDLE_HOME>
   ```

   ```
   docker build --no-cache -t "<IMAGE_NAME>:latest" .
   ```

   NOTE: If Transformation Advisor detected application dependencies it expects there will be files in the 'lib' directory to be copied. If these files are unnecessary this command may fail to run with an error similar to "No source files were specified". In this case remove the following line from the Dockerfile: ***COPY ./binary/lib/* /config/lib***

4. The base Liberty profile image will be pulled down and used to create the image that includes your migrated application

5. Run the image and confirm that it is working correctly

   ```
   docker run –p 9080:9080 <IMAGE_NAME>:latest
   ```

6. The image has now been started and mapped to the port 9080. You can access it from your browser with this link

   http:// <LIBERTY_HOME_MACHINE_IP>:9080/<APP_CONTEXT>

Useful Commands

You may wish to examine your image when it is up and running. You can do so as follows:

1. Get the <CONTAINER_ID> for your docker image that is running

   ```
   docker ps
   ```

2. Log into the container

   ```
   docker exec –ti <CONTAINER_ID> bash
   ```

Step 3: Deployment to IBM Cloud Private

In this Step you will deploy the docker image you have created to ICP and create an instance of it.

Prerequisites

1. You have completed Step 2: Containerizing Liberty
2. To complete this stage you require the following software to be installed:
   - An instance of IBM Cloud Private
   - The Kubectl CLI

You will use the following items from the Migration Bundle:

- deployment.yaml

In the following tasks:

- <MIGRATION_BUNDLE_HOME> refers to the location where you have unzipped the Transformation Advisor Migration Bundle.
- <LIBERTY_MACHINE> refers to the machine where you have installed Liberty profile
- <ICP_MACHINE> refers to the machine where you have installed IBM Cloud Private
- <ICP_INSTANCE> refers to the instance of IBM Cloud Private
- <IMAGE_NAME> is the name you use for this image, typically the name of the application.

Tasks

1. Configure your <LIBERTY_MACHINE> to connect to your <ICP_MACHINE> by adding the necessary IP and certs.

   Add the <ICP_MACHINE> IP to your hosts file, an example entry is shown below

   10.10.10.72 mycluster.icp

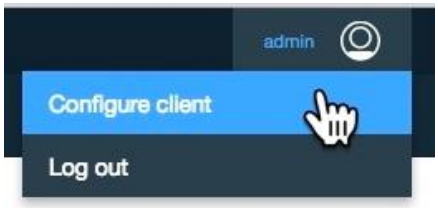
   Add the *<ICP_MACHINE> certificate to the <LIBERTY_MACHINE>*

   mkdir -p /etc/docker/certs.d/mycluster.icp:8500

   sftp into ICP and get the ca.crt from above location

2. Log into Kubernetes using the configure client option from <ICP>

- *Open your browser and log into <ICP_INSTANCE>*
- *Click on the icon for your Avatar and choose "Configure Client"*



- *Select the copy option in the dialog that appears*
- *Paste this into your <LIBERTY_MACHINE>*

Your Kubernetes has now been configured

3. Log into docker using your <ICP_INSTANCE > credentials

   docker login mycluster.icp:8500

4. Tag your image in docker so that you can push it to your image repository

   docker tag <IMAGE_NAME>:latest mycluster.icp:8500/default/<IMAGE_NAME>:latest

   NOTE: 'default' is the namespace in ICP. If you have created your own namespace you should use that.

5. Push the tagged image into the ICP image repository

   docker push mycluster.icp:8500/default/<IMAGE_NAME>:latest

6. View the image in the ICP image repository

   Log into <ICP> and select "Catalog > Images" from the navigation menu on the left

   You will see your image in the list

7. Deploy an instance of this image

   cd <MIGRATION_BUNDLE_HOME>/manifests

   kubectl apply -f deployment.yaml

Useful Commands

You may need to investigate or debug your image tagging and deployment. The commands listed here may be useful to do this.

1. To debug image start up you can identify the pods and the describe them to see any errors

   ```
   kubectl get pods
   ```

   ```
   kubectl describe pods <podIdFromAbove>
   ```

2. You can expose a port on your service after it is deployed, if you do not want to do so at deployment time. In this example port 9077 is exposed.

   ```
   kubectl expose deployment/plants --type="NodePort" --port 9077
   ```

Step 4: Deployment of Helm Charts to IBM Cloud Private

In this step you will package and deploy a Helm Chart to ICP and see it in the ICP catalog.

Prerequisites

1. You have completed Step 3: Deployment to IBM Cloud Private
2. To complete this stage you require the following software to be installed:
    - An instance of IBM Cloud Private
    - The Helm CLI
    - The IBM Cloud CLI (download here)
    - IBM Cloud Private CLI (download here)

You will use the following items from the Migration Bundle:

- The Helm bundle
    - <APPLICATION_NAME>v1.0.0.tgz

In the following tasks:

- <MIGRATION_BUNDLE_HOME> refers to the location where you have unzipped the Transformation Advisor Migration Bundle.
- <APPLICATION_NAME> the name of your application in the migration bundle
- <ICP_INSTANCE> refers to the instance of IBM Cloud Private
- <IMAGE_NAME> is the name you use for this image, typically the name of the application.

Tasks

1. Go to your Migration Bundle and the charts location

```
cd <MIGRATION_BUNDLE_HOME>
cd charts
```

2. Create the Helm Chart package

```
helm package <APPLICATION_NAME>
```

NOTE: This will produce a file called <APPLICATION_NAME>.tgz

3. Login with the IBM Cloud Private CLI

```
bx pr login -a <ICP_INSTANCE> --skip-ssl-validation
```

NOTE: Make sure you have your Kubernetes client configured to the ICP from the previous stage

4. Deploy the Helm Chart package to ICP

```
bx pr load-helm-chart --archive <APPLICATION_NAME>.tgz
```

5. Go to ICP and sync the Helm Repository
   - *Open your browser and log into <ICP_INSTANCE>*
   - *Select 'Manage' > 'Helm Repositories'*
   - *Click 'Sync repositories'*

6. Go to the Catalog and view your Helm Chart
   - *Select 'Catalog' > 'Helm Charts'*

   You will see your Helm Chart and be able to deploy instances from it

7. Deploy an instance
   - *Select your Helm Chart and click 'Configure'*

   NOTE: If you still have the instance from Step 3 you may wish to delete it first, or change the default ports being used

Useful Commands

You may wish to remove a Helm Chart from ICP

1. Remove a deployed Helm Chart

```
bx pr delete-helm-chart --name <APPLICATION_NAME> --version latest
```